**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

| | | | |
|---|---|---|---|
| Application No.: 10/824,092 | § | | |
| Filed: April 14, 2004 | § | Examiner: | Nguyen, Cindy |
| Inventor: | § | Group/Art Unit: | 2161 |
| Ajay Kumar | § | Atty. Dkt. No: | 5681-72300 |
| Title: CLASS STRUCTURE BASED ENHANCER FOR DATA OBJECT | § | | |

**PRE-APPEAL BRIEF REQUEST FOR REVIEW**

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir/Madam:

Applicant requests review of the rejection in the above-identified application. Claims 1-20 are pending in the application. For brevity, only the primary arguments with respect to independent claims are presented; additional arguments, such as those regarding other dependent claims, may be presented if and when the case proceeds to Appeal.

The Examiner rejected claims 1-39 under 35 U.S.C. § 103(a) as being unpatentable over Calusinski (U.S. Published Application 2005/0071342) (hereinafter "Calusinski") in view of Mullins et al. (U.S. Published Application 2004/0123048). Applicant respectfully traverses this rejection for at least the following reasons.

In regard to claim 1, Calusinski in view of Mullins fails to teach or suggest a class structure based data object enhancer configured to generate one or more enhanced classes corresponding to the one or more classes such that an object of the one or more enhanced classes is enhanced to persist data of the data fields to be persisted according to the persistence structure, wherein said data of the data fields to be persisted is data of said object, wherein the generation of each of said one or more enhanced classes comprises adding to the corresponding one of said one or more classes, one or more calls to persist data fields as specified by the persistence structure.

Calusinski teaches a persistence framework (*see e.g.,* persistence framework 154 of FIG. 1 of Calusinski) configured to "transform" data values from a business object of a client program (*see e.g.,* business object 102 and client program 110 of FIG. 1 of Calusinski) to a data object of the persistence framework (*see e.g.*, data object 138 and Abstract of Calusinski). This "transformation" may "[i]n simple mappings" include "mapping all fields in the business object to corresponding fields in a data object

associated with the same field name, or with synonymous field names, in the persistent data structure" (*see e.g.,* paragraph [0065] of Calusinski). Such transformation may "[i]n more complex mappings" include "converting the data values for all fields in the business object from the field name and data type of the business object to the corresponding field name and data type of the data object" (*see e.g.,* paragraph [0065] of Calusinski).

In addition to "transforming" data values from a business object of a client program to a data object of a persistence framework, Calusinski's system may also process "instruction[s] advising the framework what to do with the data in the business object" (*see e.g.,* paragraph [0044] of Calusinski). Calusinski describes examples of processing instructions as including "instructions to add the data from the business object to a new record in the persistent data store, to update an existing record in the persistent data store to conform with the data values in the business object, to delete from the persistent data store a record identified by data values in the business object, or to execute a query on the persistent data store according to data values in the business object" (*see e.g.,* Calusinski: paragraph [0044] and persistent data store 130 of Figure 1).

Note that Calusinski fails to teach anything at all about "wherein the generation of each of said one or more enhanced classes comprises adding to the corresponding one of said one or more classes, one or more calls to persist data fields as specified by the persistence structure" as recited in Applicant's claim. In addition, the Examiner acknowledges that Calusinski does not disclose this limitation (*see e.g.,* page 3 of the Final Office Action mailed December 31, 2008). The Examiner relies on the teachings of Mullins to teach the aforesaid limitation. More specifically, the Examiner cites paragraphs [0054], [0055], [0164], [0188] and [0194] of Mullins to teach the aforesaid claim limitation. However, as described in more detail below, Mullins clearly fails to overcome the deficiencies of Calusinski in this regard.

**Mullins (even when considered in combination with Calusinski) fails to teach or suggest "wherein the generation of each of said one or more enhanced classes comprises <u>adding to the corresponding one of said one or more classes, one or more calls to persist data fields as specified by the persistence structure</u>" as recited in Applicant's claim.** The cited portions of Mullins (much less any other portion of Mullins) fail to mention anything at all about "adding to the corresponding one of said one or more classes, one or more calls to persist data fields as specified by the persistence structure," much less generating one or more enhanced classes in this manner. It appears the Examiner is relying on the "CocoBase plug-ins" of Mullins to teach the generation of one or more enhanced classes according to the limitations of claim 1. For instance, in paragraph [0164], Mullins teaches:

> The mapping server of CocoBase can be extended through custom developed classes called plug-ins. A CocoBase plug-in implements the Java interface thought.CocoBase.CocoServerPlugin. This is a standard Java interface, and any Java class, which implements it and provides a public default constructor can be loaded automatically by the server at startup. Examples of such plug-ins are the cache and connection pool that ship as source code in the demos directory of the CocoBase software tools suite. (emphasis added)

While the Examiner did not provide a clear explanation as to how she is relying on the teachings of Mullins, it appears that the Examiner is somehow relying on a Java class "implement[ing] the Java interface thought.CocoBase.CocoServerPlugin" to teach "wherein the generation of each of said one or more enhanced classes comprises adding to the corresponding one of said one or more classes, one or more calls to persist data fields as specified by the persistence structure" as recited in Applicant's claim. For multiple reasons, the features of Mullin cited by the Examiner are not commensurate with the limitations of Applicant's claim.

**First**, the "the Java interface thought.CocoBase.CocoServerPlugin" is not the same as "one or more calls to persist data fields as specified by the persistence structure" as recited in Applicant's claim. Instead, "the Java interface thought.CocoBase.CocoServerPlugin" is standard Java interface (the inclusion of which defines a class as a "plug-in" according to paragraph [0164] of Mullins), not a call to persist data fields as specified by a persistent structure.

**Second,** while the "plug-ins" described by Mullins may include a "cache" plug-in, the data that is cached in Mullins system is not data of an object of a cache plug-in. Claim 1 specifically recites "wherein said data of the data fields to be persisted is data of said object" (where "said object" is an object of the one or more enhanced classes). Accordingly, for the Examiner to rely on a cache plug-in class of Mullins (*see e.g.,* paragraph [0164] of Mullins) to teach the generation of an enhanced class, Mullins would have to teach that data of an object of such a cache plug-in class is persisted according to a persistence structure. **Clearly, Mullins teaches no such thing.** In paragraph [0194], Mullins does teach "an 'in memory' cache list of objects"; however, such objects are not objects of the cache plug in class described in paragraph [0164] of Mullins.

The various other cited portions of Mullins do not overcome the deficiencies of Calusinski. For instance, in paragraph [0188], Mullins describes modifying caching features via a "cache configuration file." Note that the "cache configuration file" is not a class, and modifying such a file has nothing to do with modifying a class. Accordingly, modifying such a configuration file is clearly different than "wherein the generation of each of said one or more enhanced classes comprises adding to the corresponding one of said one or more *classes*, one or more calls to persist data fields as specified by the

persistence structure" as recited in Applicant's claim. In paragraphs [0054]-[0055], Mullins describes the composition of a "repository" and "XML files," neither of which pertain to the generation of an enhanced class, much less the generation of an enhanced class according to the specific limitations of claim 1.

**Even were the teachings of Calusinski and Mullins combined, the resultant system would not meet the specific limitations of claim 1.** More specifically, such a combination would at best result in a system where Calusinski's persistence framework (*see e.g.,* persistence framework 154 of Calusinski) were equipped with a cache plug-in and associated cache as taught by Mullins. However, that resultant system would not include the generation of one or more enhanced classes, much less "wherein the generation of each of said one or more enhanced classes comprises adding to the corresponding one of said one or more classes, one or more calls to persist data fields as specified by the persistence structure" as recited in Applicant's claim. **At best, the only "call" of the cited references that could be considered tangentially analogous to "one or more calls to persist data fields" are calls 172 and/or 174 described by Calusinski in paragraphs [0044]-[0045]; however, Calusinski clearly teaches that such calls are not added to a class and instead are *issued by client program 110*.** Since combining the teachings of Mullins with the teachings of Calusinski would not alter the manner in which client program 110 issues calls 172 and 174 and since the cited art clearly does not teach adding such calls to a class as part of generating an enhanced, the combination of the cited references cannot teach or suggest all the limitations of Applicant's claim.

**Furthermore, Applicant asserts that the Examiner has failed to provide a proper reason as to why one of ordinary skill in the art would combine the cited references in such a way that would result in Applicant's claimed invention.** More specifically, the Examiner asserts that such a combination would have been obvious "to avoid the need for object lookup again even with multiple clients" and cites paragraph [0183] of Mullins. The feature of Mullins that provides for "avoid[ing] the need for object look-up" is the placement of an object "in memory for future reference." However, this aspect is already taught by the teachings of Calusinski alone. For instance, Calusinski teaches that data object 138 is stored within persistence framework 154 (*see e.g.,* Figure 1) and that multiple actions can be performed on the data object while stored in the persistence framework 154 (e.g., *see e.g.,* transformations, updates, etc. described with respect to paragraphs [0065] and [0044] of Calusinski). Accordingly, one of ordinary skill in the art would have absolutely no need to modify Calusinski with the teachings of Mullins "to avoid the need for object lookup again even with multiple client." Instead, one of ordinary skill in the art seeking "to avoid the need for object lookup again even with multiple client" would simply use the teachings of Calusinski alone, not combine the teachings of Calusinski with the teachings of Mullins or any other reference. Moreover, "avoiding the need for lookup" has absolutely

nothing to do with enhancing a class to include calls to persist data of an object of that class. Thus, the Examiner has not provided a reason to combine the references in a way that would result in Applicant's claimed invention.

Thus, for at least the reasons presented above, the rejection of claim 1 is unsupported by the cited art and removal thereof is respectfully requested. Remarks similar to those presented above apply to claims 14 and 27.

Applicants also assert that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the rejection has been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time. Applicants reserve the right to present additional arguments.

Respectfully submitted,
/Robert C. Kowert/
Robert C. Kowert, Reg. #39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: March 31, 2009